

ALGORITMO PARA LA DETECCIÓN DEL USO DE CUBREBOCAS EN IMÁGENES

Jesús Alejandro Navarro-Acosta, Mario I. Chacón-Murguía, Juan A. Ramírez-Quintana,
Laboratorio de Percepción Visual

Tecnológico Nacional de México/I.T. Chihuahua
Av. Tecnológico #2909 Chihuahua, Chih. México CP31310
Tel. +52 (614) 2-01-2000 Fax +52 (614) 4-13-5187

M20061553@chihuahua.tecnm.mx, mario.cm@chihuahua.tecnm.mx, j.rq@chihuahua.tecnm.mx,

RESUMEN

El virus Covid-19 ha provocado un cambio histórico en el mundo entero, y ha sido necesario emplear medidas preventivas para evitar contagios, como el uso de cubrebocas. En este proyecto se propone un sistema de visión para la detección del uso de cubrebocas. Tal sistema puede detectar múltiples sujetos en una imagen y es capaz de identificar el uso de cubrebocas con distintos colores y/o texturas. El sistema consiste en dos etapas: extracción de características y clasificación. La primera se divide en dos bloques: un detector de rostros basado en CNNs; y un bloque para el análisis de características del rostro. El clasificador consiste en un discriminante lineal utilizando máquinas de vectores de soporte (SVM). El sistema obtuvo un nivel de exactitud del 93.53% en detección de uso de cubrebocas.

Palabras clave—Covid-19, cubrebocas, visión por computadora, SVM.

ABSTRACT

The Covid-19 virus has brought a historical world change, and it has become necessary to apply several preemptive actions to avoid the spread of this virus, such as facemask usage. This paper presents the development of a computer vision system for facemask detection. It can detect multiple people on a single frame, and in most cases, it can identify facemask usage regardless of the mask color and/or texture. The system is divided into two main stages: feature extraction, and classification. The first stage consists of a face detector block, using a CNN model, and a face feature analysis block. The classification stage is an SVM linear model. The proposed system achieved an accuracy of 93.53% in facemask usage recognition.

Keywords---Covid-19, facemask, computer vision, SVM.

1. INTRODUCCIÓN

En el mes de marzo del año 2020 se declaró un estado de pandemia debido a la enfermedad causada por Covid-19. Debido a la forma de propagación de tal virus, el uso de cubrebocas se ha convertido en una de las medidas preventivas más importantes. El monitoreo de tales medidas es una ardua tarea para ser llevada a cabo por un ser humano, especialmente en lugares públicos, en donde la cantidad de personas es alta y el cumplimiento de las medidas preventivas es vital para evitar contagios masivos. Debido a esto se han generado sistemas para la automatización del monitoreo de medidas preventivas, como sistemas de acceso con sensores de temperatura y dispensadores electrónicos de gel desinfectante. Durante el año 2020 se desarrollaron varios sistemas de visión para la

detección de uso de cubrebocas en diversos ambientes. En [1] se desarrolla un sistema para el análisis de imágenes de supermercados y la detección de cubrebocas en clientes y personal, en [2] y [3] se desarrollan sistemas con el mismo propósito, pero utilizando secuencias de video en ambientes urbanos. La mayoría de los algoritmos para la detección de uso de cubrebocas se basan en técnicas de aprendizaje de máquina y aprendizaje profundo [3] - [4]. En [4] se utiliza una red neuronal pre-entrenada, denominada *InceptionV3*, para identificar personas en la imagen y se añaden capas adicionales para la detección de uso de cubrebocas. En [5] se desarrolló el sistema para la detección de cubrebocas a partir de la red neuronal convolucional *YOLOv3*. Un reto en la detección de cubrebocas es la gran variedad de colores, estilos y formas que estos pueden tener, en [6] se propone un modelo que consiste en dos etapas: la primera corresponde a una red profunda para localizar a la persona o personas presentes en la imagen; y la segunda etapa consiste en la extracción de características de la región del rostro de la persona, utilizando un análisis de color. En este artículo se propone un algoritmo para la detección del uso de cubrebocas (ADUC) basado en el trabajo de Buciu [6], para el cual se propusieron dos mejoras al proceso de extracción de características. Y se utilizó un modelo clasificador lineal utilizando máquinas de vectores de soporte (SVM, por sus siglas en inglés).

2. DESARROLLO DE ALGORITMO PARA LA DETECCIÓN DEL USO DE CUBREBOCAS

El sistema propuesto, denominado ADUC, se compone por dos etapas principales: extracción de características y clasificación. La primera etapa se divide en dos bloques: el primero, denominado BDR, se encarga de la detección de rostros en una imagen dada para extraer regiones de interés; y el segundo bloque, denominado BARI, se encarga de analizar las regiones de interés para generar un vector de características. La segunda etapa consiste en la clasificación del vector de características para determinar si la región de interés corresponde a una persona utilizando cubrebocas o no, esta etapa consiste en un único bloque denominado BDC. En la Figura 1 se muestra el diagrama a bloques del sistema.

En las siguientes secciones se describe a detalle el funcionamiento de cada uno de los bloques del sistema ADUC.

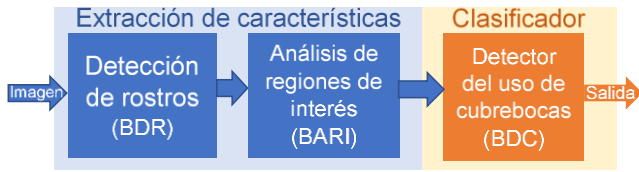


Figura 1. Diagrama general del sistema ADUC.

2.1. Etapa de extracción de características de los rostros en la imagen

Como se mencionó anteriormente, el objetivo de esta etapa es el de identificar regiones de interés, correspondientes a rostros humanos, y generar un vector de características por cada región presente en la imagen de entrada.

2.1.1. Detección de rostros

La ecuación (1) describe de manera general el propósito de este bloque.

$$I(x, y) \xrightarrow{\text{Det. Rostros}} \left\{ \begin{array}{l} I_{F_i}(x, y) \\ L_i \end{array} \right\} \forall i = 1, 2, \dots, n \quad (1)$$

En donde $I(x, y)$ corresponde a la imagen de entrada en el espacio de color RGB; $I_{F_i}(x, y)$ se refiere a la i -ésima sub-imagen correspondiente a un rostro de la imagen; L_i es un arreglo que contiene información sobre la posición de ojos, nariz y boca del i -ésimo rostro, este se compone de 5 pares de coordenadas (ojo izquierdo, ojo derecho, nariz, borde derecho de boca y borde izquierdo de boca); y n es el total de rostros en la escena. Para identificar las regiones de interés, se utilizó el modelo para la detección de rostros MTCNN [7], el cual consiste en tres etapas de redes neuronales convolucionales (CNN): la primera se denomina P -net, esta red se encarga de determinar las posibles regiones de interés de la imagen de entrada; la segunda se denomina R -net, y su propósito es eliminar aquellas regiones de interés que no corresponden a un rostro humano; finalmente, la última etapa, denominada O -net, tiene como objetivo identificar la posición de ojos, nariz y boca dentro de la región del rostro. Para la implementación se utilizó la librería *mtcnn* de acceso libre para lenguaje Python. En la Figura 2 se muestra un ejemplo de la detección de rostros utilizando este modelo.

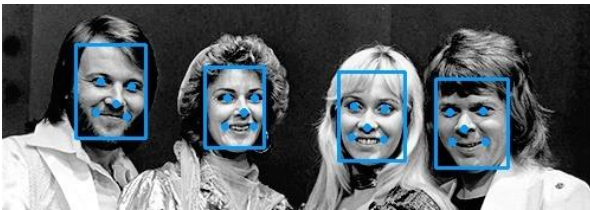


Figura 2. Detección de rostros utilizando modelo MTCNN.

2.1.2. Análisis de rostro

En este bloque se genera un vector de características para cada rostro detectado. Tal vector contiene la información necesaria para la detección del uso de cubrebocas.

2.1.2.1. Método cociente de color

Como punto de partida se tomó el método propuesto en [6], el cual consiste en el cálculo de un vector de características con dos cocientes de color. El procedimiento es el siguiente:

- A. División de sub-imagen correspondiente a rostro en región superior e inferior.

$$I_{U_i}(x, y) = I_{F_i}(x, y) \quad 1 \leq y \leq 0.5H \quad (2)$$

$$I_{L_i}(x, y - 0.5H) = I_{F_i}(x, y) \quad 0.5H < y \leq H \quad (3)$$

En donde $I_{U_i}(x, y)$ e $I_{L_i}(x, y)$ son las regiones superior e inferior, respectivamente, de la imagen del rostro $I_{F_i}(x, y)$; y H es la dimensión de $I_{F_i}(x, y)$ a lo largo del eje vertical. Es decir, la división se realiza justo a la mitad de la sub-imagen del rostro.

- B. Se determina la media de los canales RGB para cada región

$$mR_{U_i} = E[I_{U_{iR}}(x, y)] \quad mR_{L_i} = E[I_{L_{iR}}(x, y)]$$

$$mG_{U_i} = E[I_{U_{iG}}(x, y)] \quad mG_{L_i} = E[I_{L_{iG}}(x, y)] \quad (4)$$

$$mB_{U_i} = E[I_{U_{iB}}(x, y)] \quad mB_{L_i} = E[I_{L_{iB}}(x, y)]$$

En donde mR_{U_i} , mG_{U_i} y mB_{U_i} son, respectivamente, la media de los canales rojo, verde y azul de la región superior del rostro $I_{U_i}(x, y)$; y mR_{L_i} , mG_{L_i} y mB_{L_i} corresponden a los valores esperados de cada canal para la región inferior del rostro $I_{L_i}(x, y)$.

- C. Se genera el vector de características de cociente de color (CQF , por sus siglas en inglés).

Para ello, se calcula la relación entre la media de los canales rojo-verde y rojo-azul para las regiones superior e inferior:

$$rg_{L_i} = \frac{mR_{L_i}}{mG_{L_i}} \quad rg_{U_i} = \frac{mR_{U_i}}{mG_{U_i}} \quad rb_{L_i} = \frac{mR_{L_i}}{mB_{L_i}} \quad rb_{U_i} = \frac{mR_{U_i}}{mB_{U_i}} \quad (5)$$

Después, se calculan dos cocientes, los cuales indican la relación de los canales rojo-verde y rojo-azul, respectivamente, entre las regiones superior e inferior:

$$rgq_i = \frac{rg_{L_i}}{rg_{U_i}} \quad rbq_i = \frac{rb_{L_i}}{rb_{U_i}} \quad (6)$$

Finalmente, con estos valores se genera el vector CQF :

$$CQF_i = [rgq_i \quad rbq_i] \quad (7)$$

Estas métricas fueron planteadas bajo la hipótesis de que, para un rostro descubierto, los valores de rgq_i y rbq_i tienden a ser iguales o muy cercanos a 1, ya que tanto en $I_U(x, y)$ como en $I_L(x, y)$ predomina el color correspondiente a la piel, de lo contrario, si tales valores son distintos a 1, podría indicar que existe una oclusión en el rostro, como un cubrebocas.

2.1.2.2. Modificaciones al método cociente de color

Se propusieron dos mejoras al método de cociente de color: un nuevo criterio para la selección de las regiones superior e inferior del rostro; y un elemento adicional al vector de características utilizando la relación de los canales verde-azul. El nuevo criterio de selección de regiones de interés consiste en extraer tres porciones del rostro, con base en la ubicación de ojos, nariz y boca. Tales regiones son: entre ojos $I_{BE}(x,y)$; frente $I_{FH}(x,y)$; y entre nariz y boca $I_{NM}(x,y)$. En la Figura 3 se muestra la posición de las regiones mencionadas.

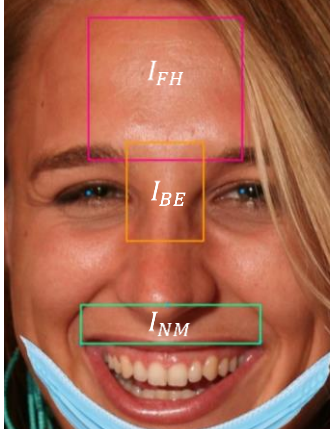


Figura 3. Regiones de interés dentro de la sub-imagen de correspondiente a un rostro.

A la región inferior del rostro $I_L(x,y)$, para todos los casos, se asigna la región $I_{NM}(x,y)$:

$$I_L(x,y) = I_{NM}(x,y) \quad (8)$$

Por otro lado, para seleccionar la región superior, se elige la región más uniforme entre las regiones $I_{BE}(x,y)$ e $I_{FH}(x,y)$, es decir, aquella en donde la mayoría de los píxeles correspondan únicamente a la piel. Para esto, se calcula la desviación estándar (σ) de los valores de ambas regiones y se asigna a $I_U(x,y)$, aquella región con una σ menor:

$$I_U(x,y) = \begin{cases} I_{FH}(x,y) & \text{si } \sigma_{FH} < \sigma_{BE} \\ I_{BE}(x,y) & \text{de otra forma} \end{cases} \quad (9)$$

Mediante el uso de las regiones $I_{BE}(x,y)$, $I_{FH}(x,y)$ e $I_{NM}(x,y)$ se pretende que la información de $I_U(x,y)$ e $I_L(x,y)$ sea más útil respecto al criterio de selección descrito por las ecuaciones (2) y (3). Con el criterio original, diversos factores, como el color de ojos, peinado de la persona, o el uso de anteojos, entre otros, influyen directamente en el valor de los elementos del vector CQF . Por otro lado, al seleccionar áreas específicas dentro del rostro, se reduce el efecto de tales factores en el vector CQF , como se puede observar en la Figura 3, en donde la mayor parte de las regiones $I_{BE}(x,y)$, $I_{FH}(x,y)$ e $I_{NM}(x,y)$ corresponden únicamente a porciones de piel del rostro.

La segunda mejora consiste en el cálculo de un nuevo elemento para el vector de características. Tal elemento corresponde a un

tercer cociente de color para indicar la relación de los canales verde-azul entre las regiones superior e inferior del rostro. A partir de las expresiones mostradas en las ecuaciones (5) y (6), se calcula el tercer elemento de la siguiente manera:

$$gb_{L_i} = \frac{mG_{L_i}}{mB_{L_i}} \quad gb_{U_i} = \frac{mG_{U_i}}{mB_{U_i}} \quad (10)$$

En donde gb_L es la relación entre las medias de los canales verde y azul de la región superior del rostro, y gb_U corresponde a la misma relación para la región inferior. Finalmente, se calcula el cociente gbq_i :

$$gbq_i = \frac{gb_{L_i}}{gb_{U_i}} \quad (11)$$

En [6] se utilizaron los elementos rgq y rbq tomando en cuenta que el color de la piel está correlacionado principalmente con los valores de los píxeles en el canal rojo de la imagen, sin embargo, como se demostrará en la siguiente sección, tales variables presentan una alta correlación entre sí. En este caso, el elemento gbq , al contener información respecto a los canales verde y azul, se relaciona más a colores que no corresponden a la piel de la persona, de manera que presenta una menor correlación respecto a rgq y rbq .

2.1.2.3. Análisis de vector de características

Luego de generar el algoritmo de extracción de características, se analizó una porción de las bases de datos utilizadas y se generó el vector de características CQF correspondiente a cada uno de los rostros detectados. Después, se calculó la matriz de correlación, mostrada en la Tabla 1, para evaluar el grado de similitud entre las variables.

Tabla 1. Matriz de correlación CQF .

	rgq	rbq	gbq
rgq	1	0.9698	0.7862
rbq	0.9698	1	0.9
gbq	0.7862	0.9	1

Se determinó que la variable rbq está altamente relacionada a rgq . Por otro lado, las variables rgq y gbq tienen el menor valor de correlación en la matriz, lo cual sugiere que este par de variables serían más aptas para generar un vector de características de dos variables. Con base en lo anterior, se propuso un nuevo vector de características $MCQF$:

$$MCQF_i = [rgq_i \quad gbq_i] \quad (12)$$

2.2. Etapa de clasificación

Esta etapa consiste en un bloque denominado BDC y tiene como objetivo determinar la clase a la que pertenece el vector

de características *MCQF*. En este caso se trata de una clasificación binaria: rostro con cubrebocas y rostro sin cubrebocas.

2.2.1. Datos de entrenamiento y prueba

Se utilizaron muestras de tres bases de datos:

Face mask dataset (FMD) [8]: se compone por 853 imágenes reales de personas utilizando cubrebocas y personas sin cubrebocas en distintos escenarios y condiciones. La mayoría contiene más de un rostro y los cubrebocas tienen diversos estilos. Para este proyecto se seleccionaron 628 imágenes de FMD, las cuales corresponden a muestras con un máximo de 5 rostros. Este límite se estableció ya que, en la mayoría de las muestras en donde existen más de 5 objetivos, el tamaño y calidad de las regiones correspondientes a rostros disminuyen considerablemente.

MaskedFace-Net (MFN) [9]: compuesta por imágenes artificiales con una sola persona utilizando un estilo de cubrebocas. Las imágenes corresponden a muestras de la base de datos FFHQ que fueron editadas para añadir el cubrebocas. Se seleccionaron al azar 300 imágenes de esta base de datos. Se optó por utilizar menos muestras de esta base de datos respecto a las de FMD ya que en MFN el cubrebocas únicamente tiene un estilo y color.

CelebA [10]: consiste en imágenes de celebridades. Se eligieron al azar 500 imágenes de esta base de datos, las cuales corresponden a la clase de rostros sin cubrebocas. Se utilizaron estas muestras para evitar un desbalance entre las clases.

En total, se utilizaron 1428 imágenes. Las cuales, como generalmente se suele llevar a cabo en modelos de aprendizaje de máquina [11], se dividieron en dos conjuntos: entrenamiento y prueba, correspondiente a un 80% y 20%, respectivamente, de las imágenes de cada base de datos.

2.2.2. Análisis de espacio de características y diseño de clasificador

Dado que el vector de características se compone por dos variables, es posible visualizar los datos en el espacio de características. En la Figura 4 se muestra la distribución de los datos de entrenamiento en el espacio de características. Los datos de la clase de rostros utilizando cubrebocas tienen una mayor dispersión respecto a los datos de la clase de rostros sin cubrebocas. En la Tabla 2 se muestra la media y desviación estándar de cada variable para los datos de ambas clases. Nótese que el comportamiento de los datos es acorde a la hipótesis planteada, las muestras correspondientes a los rostros sin cubrebocas tienen un vector de características cuyos valores de *rgq* y *gbq* son cercanos a 1, por otro lado, para los rostros con cubrebocas, tales valores tienden a ser menores a 1.

A partir del análisis del espacio de características, se optó por utilizar un clasificador lineal para llevar a cabo la detección del uso de cubrebocas. De manera que es necesario encontrar una función discriminante lineal (LDA, por sus siglas en inglés) que

se adapte a los datos de entrenamiento. La ecuación (13) corresponde al LDA.

$$g(MCQF) = \omega^T MCQF + \omega_0 \quad (13)$$

En donde ω y ω_0 corresponden a los parámetros de la función lineal, también denominados pesos.

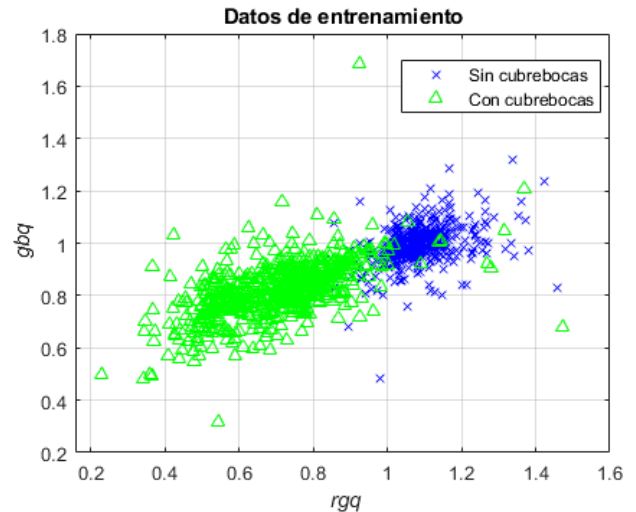


Figura 4. Distribución de los datos de entrenamiento en el espacio de características.

Tabla 2. Media y desviación estándar de las variables *rgq* y *gbq* para las clases de rostros con cubrebocas y rostros sin cubrebocas.

	Media / Desviación estándar	
	Con cubrebocas	Sin cubrebocas
<i>rgq</i>	0.7097 / 0.1525	1.0782 / 0.0819
<i>gbq</i>	0.8211 / 0.1059	0.9925 / 0.0700

Para encontrar los pesos de tal clasificador se utilizó el método de máquinas de vectores de soporte (SVM, por sus siglas en inglés). Se utilizaron herramientas de software para encontrar los vectores de soporte y llevar a cabo el proceso de entrenamiento, específicamente de la librería *scikit-learn* para el lenguaje de programación *Python*.

3. RESULTADOS

A continuación, se presentan los resultados de la implementación del sistema. Todos los algoritmos fueron desarrollados con el lenguaje de programación *Python 3.8*, y se utilizaron diversas librerías de código abierto para el procesamiento de imágenes (*openCV*) y herramientas matemáticas (*NumPy*, *SciPy*, *Matplotlib*). Se utilizó un equipo de cómputo con un procesador Intel® Core™ i7-4510U.

3.1. Detección de rostros

Para calcular el desempeño del bloque para la detección de rostros se analizó la cantidad de rostros detectados

correctamente, los falsos positivos (número de detecciones que no corresponden a un rostro) y el número real de rostros presentes en las imágenes o *ground truth* (GT). Para el caso de las bases de datos *MaskedFace-Net* y *CelebA*, el total de objetivos es 1 para cada una de las muestras y el tamaño del rostro es similar en todos los datos, de manera que la comprobación de aciertos se llevó a cabo verificando que la cantidad de rostros detectados por imagen y el tamaño de cada región. Por otro lado, para la base de datos FMD, el número de rostros por imagen varía. Para conocer los aciertos, se comparó el número y la posición de cada rostro detectado con el GT de cada una de las imágenes. Para la evaluación: se considera acierto (VP, verdadero positivo) si el rostro detectado coincide con un objetivo esperado del GT; se considera error (FN, falso negativo) cada objetivo esperado del GT que no haya sido detectado; si se detecta un rostro que no coincida con un objetivo esperado del GT, se considera como falso positivo (FP). Nótese que, para este caso, los verdaderos negativos (VN) no son contables, puesto que existe un número infinito de regiones en una imagen que no corresponden a un rostro, por lo que se omite tal categoría. En la Tabla 3 se muestra el desempeño del detector de rostros para cada base de datos utilizada. En la Figura 5 se muestran algunos ejemplos de imágenes cuyos rostros no fueron detectados correctamente.

Tabla 3. Precisión y exhaustividad (*recall*) del detector de rostros para cada base de datos.

Base de datos	VP	FN	FP	Precisión de detección de rostros	Recall de detección de rostros
FMD	796	243	57	0.9331	0.7661
MFN	289	11	16	0.9475	0.9633
CelebA	498	2	3	0.9940	0.9960



Figura 5. Ejemplos de muestras en donde no se reconoció el total de rostros correctamente.

También se verificó el tiempo de procesamiento del detector de rostros utilizando imágenes de distintas resoluciones. Los resultados se muestran en la Tabla 4. Nótese que, al aumentar el número de rostros en la imagen, el tiempo de procesamiento incrementa ligeramente. Sin embargo, los factores principales

que influyen en el tiempo de procesamiento son la resolución de la imagen y la plataforma de hardware.

Tabla 4. Tiempo de procesamiento de algoritmo MTCNN para la detección de rostros en imágenes

Resolución de imagen	Tiempo de procesamiento detección de rostros	
	Imagen con 1 rostro	Imagen con 8 rostros
178x218	0.4975 s	0.5299 s
640x480	0.7512 s	0.8170 s
1024x1024	1.0978 s	1.2287 s

3.2. Detección de uso de cubrebocas

Como se mencionó anteriormente, se utilizó un SVM para generar el modelo del clasificador lineal del sistema. Los pesos finales y la expresión $g(MCQF)$ correspondiente al LDA se muestran en (14).

$$\omega = [-9.102 \quad -2.209], \quad \omega_0 = 10.502 \quad (14)$$

$$g(MCQF) = -9.102rgq - 2.209gbq + 10.502$$

En la Figura 6 se muestra, de manera gráfica, el resultado de la prueba del sistema y se indica la división entre clases a partir de la línea recta generada por la expresión correspondiente al LDA en $g(MCQF)=0$.

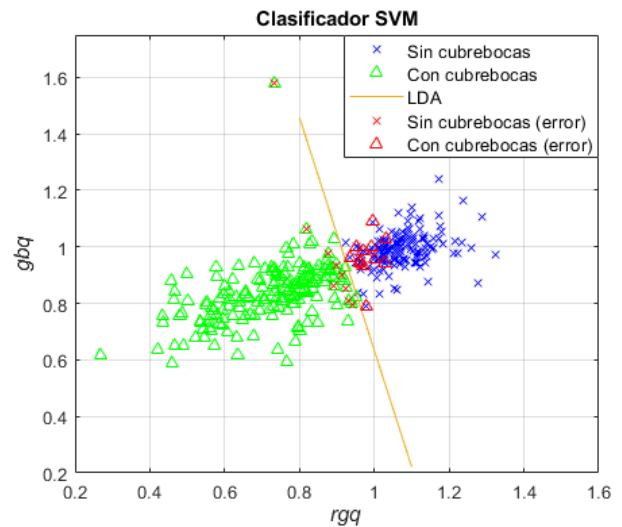


Figura 6. Representación gráfica de clasificador lineal y resultados con datos de prueba.

Al realizar la clasificación de los datos de prueba, se obtuvieron los resultados mostrados en la Tabla 5. Para medir el desempeño del clasificador se calcularon las métricas de exactitud, precisión y exhaustividad, cuyos valores se muestran en la Tabla 6.

Tabla 5. Matriz de confusión con datos de prueba.

		Predicción	
		Rostro sin cubrebocas	Rostro con cubrebocas
Realidad	Rostro sin cubrebocas	162	9
	Rostro con cubrebocas	12	142

Tabla 6. Desempeño del clasificador con muestras de entrenamiento y prueba.

Métrica	Desempeño con datos de:	
	Entrenamiento	Prueba
Exactitud	0.96116	0.93538
Precisión	0.97327	0.94737
Exhaustividad	0.95377	0.93103

4. CONCLUSIONES

En este trabajo se describe un sistema de visión para la detección del uso de cubrebocas utilizando un detector de rostros basado en CNNs y un análisis de cociente de color de regiones específicas del rostro. El sistema propuesto obtuvo un desempeño competente respecto al modelo original propuesto en [6]. La exactitud promedio del método original fue de 0.9474 utilizando un clasificador no-lineal con un kernel de base radial, mientras que el método propuesto en este trabajo alcanzó una exactitud de 0.9353, utilizando un clasificador lineal, cuya complejidad es menor que en el modelo original. Las mejoras propuestas al método cociente de color para la extracción de características permiten que el sistema sea potencialmente más robusto ante la presencia de diversos elementos en la imagen de un rostro y otros rasgos, como color de ojos, peinado, anteojos, etc. Bajo el esquema propuesto, al seleccionar más regiones específicas dentro del rostro para la etapa de extracción de características, sería posible realizar la detección del uso incorrecto de cubrebocas en próximos trabajos.

A la fecha de conclusión de este proyecto, a lo largo del mundo se han aplicado más de 700 millones de vacunas contra el virus Covid-19, sin embargo, aún queda un largo camino por recorrer para proteger completamente a la población contra la enfermedad causada por tal virus, de manera que es posible que el uso de cubrebocas continúe siendo una medida de prevención necesaria en espacios públicos cerrados durante varios meses o incluso años.

AGRADECIMIENTO

Se agradece al Tecnológico Nacional de México/I.T. Chihuahua por brindar el apoyo para el desarrollo de este trabajo bajo el proyecto 10071.21-p.

REFERENCIAS

- [1] W. Han, Z. Huang, A. Kuerban, M. Yan y H. Fu, "A Mask Detection Method for Shoppers Under the Threat of COVID-19 Coronavirus," *2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, pp. 442-447, 2020.
- [2] G. Draughon, P. Sun y J. Lynch, "Implementation of a Computer Vision Framework for Tracking and Visualizing Face Mask Usage in Urban Environments," *2020 IEEE International Smart Cities Conference (ISC2)*, pp. 1-8, 2020.
- [3] A. S. Joshi, S. S. Joshi, G. Kanahasabai, R. Kapil y S. Gupta, "Deep Learning Framework to Detect Face Masks from Video Footage," *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 435-440, 2020.
- [4] M. S. Islam, E. Haque Moon, M. A. Shaikat y M. Jahangir Alam, "A Novel Approach to Detect Face Mask using CNN," *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, pp. 800-806, 2020.
- [5] M. R. Bhuiyan, S. A. Khushbu y M. S. Islam, "A Deep Learning Based Assistive System to Classify COVID-19 Face Mask for Human Safety with YOLOv3," *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1-5, 2020.
- [6] G. Jignesh Chowdary, N. S. Punn, S. K. Sonbhadra y S. Agarwal, "Face Mask Detection Using Transfer Learning of InceptionV3," *Big Data Analytics. BDA 2020. Lecture Notes in Computer Science*, vol. 12581, pp. 81-90, 2020.
- [7] I. Buciu, "Color quotient based mask detection," *2020 International Symposium on Electronics and Telecommunications (ISETC)*, pp. 1-4, 2020.
- [8] K. Zhang, Z. Zhang, Z. Li y Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," *IEEE Signal Processing Letters*, vol. 23, n° 10, pp. 1499-1503, 2016.
- [9] Kaggle, "Face Mask Detection," [En línea]. Available: <https://www.kaggle.com/andrewmvd/face-mask-detection>. [Último acceso: 3 Marzo 2020].
- [10] A. Cabani, K. Hammoudi, H. Benhabiles y M. Melkemi, "MaskedFaceNet – A dataset of correctly/incorrectly masked face images in the context of COVID-19," *Smart Health*, vol. 19, 2021.
- [11] Z. Liu, P. Luo, X. Wang y X. Tang, "Deep Learning Face Attributes in the Wild," de *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.
- [12] Google Inc, "Machine Learning Crash Course," 10 Febrero 2020. [En línea]. Available: <https://developers.google.com/machine-learning/crash-course/training-and-test-sets/splitting-data>. [Último acceso: 23 Junio 2021].